Exhibit 1

Home > Products > Software >

Free and Open Source Java

Overview About Java Community FAQ Get Involved

FAQ

The Announcement

Q: What is Sun announcing?

A: Sun is open sourcing all of its Java platform implementations under same license (called GPL Version 2) used by the GNU/Linux operating system.

Specifically Sun is announcing:

- GPL v2 license for Sun's Java SE (JDK) and Java ME implementations, and adding this license to Sun's Java EE implementation.
- First release of code for the JDK and for Sun's Java ME implementation, projects and communities.
- Roadmap for future code releases and community development.

This singular act is the largest contribution ever made to the free software community, and places Sun squarely at the front of the open-source movement - as the single biggest commercial contributor.

Source

- OpenJDK
- Mobile & Embedded
- GlassFish

Communities

- OpenJDK
- Mobile & Embedded
- GlassFish

Related Resources

- Developers.sun.com
- NetBeans.org
- Java.net
- Java.sun.com
- Java.com
- Java Community Process

Q: What does Sun mean by "compatibility?"

A: Compatibility for a Java technology means the implementation of that technology meets the associated compatibility requirements of its Technology Compatibility Kit or TCK. For Java SE this means passing the TCK tests and other requirements defined in the Java SE TCK.

Back to top

Q: Do you think anyone will fork the JDK?

A: We expect great new ideas and valuable research to come from forks to the platform. In addition, Sun encourages compatible forks where the code is ported to additional hardware and software platforms. Such ports extend the breadth of Java SE to places currently not supported by any vendor. Broad distribution of incompatible forks is potentially a danger since such forks could damage the "Write Once, Run Anywhere" compatibility value of the Java platform.

Back to top

Q: So, what about compatibility? How will Java technology remain "Write Once, Run Anywhere" after Sun's Java platform implementations are open sourced?

A: The Java technology compatibility promise is so central to the value of the platform that it has been the single most important influence in driving the detailed planning and decisions for this initiative. Sun is making a number of key decisions and commitments to the community that we believe will help foster compatibility:

- License: GPL makes proprietary forks less likely, as all changes must be published.
- Branding: the Java trademark and logos are for compatible implementations only.
- JCP: the JCP's role as the governing body for Java standards and evolution is not changed by this move.
- Community Development: Sun's experience in building strong communities means developers will likely find the governance and infrastructure to their liking.

In addition to the specific steps that we're taking to help foster compatibility, we are convinced that the market will demand compatible implementations, and that incompatible ones won't gain traction. With the billions of dollars of Java applications in the installed base, the community has recognized that a Java implementation that doesn't run the installed base of code won't get very far. That is why both the GNU/Classpath and Apache Harmony projects have stated unequivocally that they're working to build fully compatible implementations of Java SE.

We have experience in the Java EE world to back this up: there are four compatible, open-source Java EE implementations in the market including Sun's GlassFish Application Server, and no incompatible variants have achieved any market penetration. We expect the same dynamic to happen with both Java SE and Java ME platforms.

Back to top

Business Model

Q: What do Java SE customers buy from Sun?

A: The JRE and JDK binaries themselves will remain zero cost downloads. End-user customers will buy subscriptions and one-time engagements for enterprise-class support from Sun for Java SE, including:

- High quality, optimized performance binaries released on a predictable schedule.
- Packaged updates, patches, and security fixes.
- Break/fix support including customer-specific bug fixes delivered out of cycle.
- Branding: Access to trademark and logo licenses including "Java Powered".
- Installation, custom engineering services.
- Training, education.
- Developer support.

Commercial source code licensees may optionally buy additional services and gain additional rights, including:

- The right to ship binaries derived from custom modified source code.
- The right to make modifications to the source code without putting those modifications back into the open-source code commons.
- The right to use the "Java Compatible" logo for compliant custom implementations.

LGPL instead?
- Does this mean Sun is abandoning CDDL?
- Will you license the entire JDK under this method when you open-source the rest of the code base?
- How can you ship the JDK with binary-only elements then? You said there were encumbrances.
- Why didn't you choose a license like BSD or Apache v2?
- Why is the license different for Java ME than for the JDK? Why no Classpath exception for Java ME?
- What are the advantages of GPL v2?
- What rights do developers have under GPL v2 + Classpath exception?
- What are developers' obligations under this license?
- How can someone use the open-source code base to create a derivative work?
- Can someone create an implementation that isn't compatible with the Java specification using this code?
- Can someone create software that doesn't even implement Java, but uses pieces of the JDK? What are the limitations, if any?
- Can I call the resulting software "Java"?
- What must I do to call my software based on code from the OpenJDK or phoneME projects "Java"?
- What must I do to use the "cup and steam" logo? Does Sun license it?
- How do I know which components in the OpenJDK project are under just the GPL, and which are under the GPL + Classpath exception?
- What about GPL v3? Have you considered using that license?
- I want to contribute. Do I need to sign anything to get started?
- Why do you have a Contribution Agreement?
- Do I lose any rights to my contribution under the SCA?
- But other communities are non-profits. Why should I share my copyrights with Sun?
- What can Sun do with my contribution?
- Will Sun continue to offer the JDK and Java ME source code under a commercial license, along with GPL v2?
- Why will Sun continue to offer the JDK and Java ME sources under a commercial license?
- What is the process to get a commercial license for the JDK, Java ME implementations, or Java ME TCK Framework source code?
- Will Sun continue to offer its binary bundles for the JDK and JRE?
- Is Sun changing the terms of the Binary Code License for Sun's JRE?
- What about the JCP's Spec License? Is that changing?
- Is Sun changing the terms under which it licenses the Java SE and Java ME TCKs ?
- Can I combine a virtual machine derived from the open-source Java HotSpot code and an existing Sun Java class library?
- How can Sun have other licenses that bear on the open-source code base? Isn't that no longer "open source"?
- Will Sun offer binary builds of the open-source JDK code base for download? If so, under what license?

Q: What license did you choose for the open-source JDK components?

A: GPL v2 for almost all of the virtual machine, and GPL v2 + the Classpath exception for the class libraries and those parts of the virtual machine that expose public APIs.

Back to top

Q: What license did you choose for implementations of Java ME and the Java ME frameworks?

A: GPL v2 was chosen for all components related to Java ME.

Back to top

Q: Did you use the GPL v2 verbatim, or did you alter it?

A: No alterations. Sun is using the unmodified GPL v2.

Back to top

Q: Is GPL v2 an open-source license?

A: Yes, see http://opensource.org/licenses/gpl-license.php.

Back to top

Q: What is the Classpath exception?

A: The Classpath exception was developed by the Free Software Foundation's GNU/Classpath Project (see http://www.gnu.org/software/classpath/license.html). It allows you to link an application available under any license to a library that is part of software licensed under GPL v2, without that application being subject to the GPL's requirement to be itself offered to the public under the GPL.

Back to top

Back to top

Q: Why is the license different for Java ME than for the JDK? Why no Classpath exception for Java ME?

A: With Java ME, there is no way to install and integrate implementations. Instead, implementations are integrated with the hardware. As such, the problem the Classpath exception is solving with the Java SE implementation isn't present with Java ME.

Back to top

Q: What are the advantages of GPL v2?

A: As well as fulfilling its original purpose of promoting Free software, the GPL v2 is designed to help advance projects and code commons by requiring innovation sharing with the commons. By design, it minimizes proprietary forks by requiring any modifications be shared with the project. GPL v2 is the right license to preserve Java's trademark "Write Once, Run Anywhere" value proposition.

Back to top

Q: What rights do developers have under GPL v2 + Classpath exception?

A: The best source for answering this question is the license itself (http://www.gnu.org/licenses/gpl.html). In addition, there have been numerous analyses of the GPL license and its terms, not least by the license stewards, the Free Software Foundation (http://www.fsf.org/). Sun recommends studying these interpretations, and consulting legal counsel as necessary to understand your rights under this license.

Back to top

Q: What are developers' obligations under this license?

A: Once again - the best source for understanding Sun's chosen license for the JDK source code is the license itself.

Back to top

Q: How can someone use the open-source code base to create a derivative work?

A: Sun's Java platform implementations are now Free software. Developers can use the open-source JDK and Java ME code bases in any way that a GPL v2 licensed code base may be used. These implementations are not "special cases".

Back to top

Q: Can someone create an implementation that isn't compatible with the Java specification using this code?

A: Yes. We do not recommend or endorse that action, however. In addition, they cannot label that implementation with the Java Compatible or Java Powered brand and logo. These brands are your assurance that an implementation has passed the relevant TCKs.

Back to top

Q: Can someone create software that doesn't even implement Java, but uses pieces of the JDK? What are the limitations, if any?

A: Yes. There are no limitations. But there is an obligation, as with any code licensed under GPL v2, to ensure that all resulting programs that incorporate the pieces used are accompanied by full source code as required by the license.

Back to top

Q: Can I call the resulting software "Java"?

A: No.

Back to top

Q: What must I do to call my software based on code from the OpenJDK or phoneME projects "Java"?

A: The requirements for the use of the "Java" trademark and name have not changed with the open sourcing of the JDK and Java ME source code. The GPL v2 does not include a trademark license - no OSI-approved open-source licenses do. Sun does not currently have a licensing program that permits the use of the "Java" mark in your product or company name. You can use a truthful "tagline" however associating your product or company with Java technology, according to Sun's standard terms for use for trademarks. Please see http://www.sun.com/policies/trademarks/ for more details.

Back to top